UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/714,051	11/15/2000	Stepan B. Sokolov	5181-60300	4377

7590 08/01/2003

Robert C Kowert
Conley Rose & Tayon P C
P O Box 398
Austin, TX 78767-0398

EXAMINER	
GROSS, KENNETH A	

ART UNIT	PAPER NUMBER
2122	4

DATE MAILED: 08/01/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No. 09/714,051	Applicant(s) SOKOLOV, STEPAN B.	
	Examiner Kenneth A Gross	Art Unit 2122	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☐ Responsive to communication(s) filed on _____ .

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *11-54* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *11-54* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☒ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

11)☐ The proposed drawing correction filed on _____ is: a)☐ approved b)☐ disapproved by the Examiner.

If approved, corrected drawings are required in reply to this Office action.

12)☐ The oath or declaration is objected to by the Examiner.

Priority under 35 U.S.C. §§ 119 and 120

13)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____ .

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

14)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).

a) ☐ The translation of the foreign language provisional application has been received.

15)☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449) Paper No(s) 3 .
4) ☐ Interview Summary (PTO-413) Paper No(s). _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: .

DETAILED ACTION

Specification

1. The use of the trademarks "JAVA" and "JAVASCRIPT" has been noted in this application. It should be capitalized wherever it appears and be accompanied by the generic terminology.

Although the use of trademarks is permissible in patent applications, the proprietary nature of the marks should be respected and every effort made to prevent their use in any manner which might adversely affect their validity as trademarks.

Claim Rejections - 35 USC § 112

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 19-23, 34, and 52 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, it is not clear whether the platform-independent programming language objects comprised in one or more libraries are the same platform-independent programming language objects referenced in the parent claim as storing logical commands. Clarification is requested. Similar problems can be found in corresponding Claims 34 and 52. Claims 20-23 are rejected for being dependent on a rejected parent claim.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claims 11, 13-19, 21, 23-28, 30-38, 40-46, and 48-54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang (U.S. Patent Number 6,292,936) in view of "The IR to VMx86 Translation Module Specification" by Chris Lattner, December 1999 (hereinafter Lattner).

In regard to Claim 11, Wang teaches: (a) accessing a sequence of script language instructions (Column 4, lines 43-47); (b) generating a platform-independent representation of the one or more script language instructions (Column 4, lines 48-58); (d) wherein the platform-independent programming language produces results in accordance with the original sequence of script language instructions. The interpreter inherently converts the script into an equivalent intermediate form (Column 1, lines 19-21). Wang teaches that the platform-independent programming language representation comprises a sequence of logical commands (an interpreter naturally translates the instructions into an interpreted language) but does not teach that each of the commands is stored as one or more platform-independent programming language objects. Lattner, however, does teach representing instructions using a Java 'Instruction' class (Page 2, lines 18-26). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to access a sequence of script language instructions and generate a platform-independent representation of the one or more script language instructions where the platform independent representation produces equivalent results with the script language instructions, as

taught by Wang, where the platform-independent representation includes instructions represented by objects, as taught by Lattner, since programming objects are easier to create and manipulate.

In regard to Claim 13, Lattner teaches an Operator class (Page 23). Claim 46 corresponds directly with Claim 11 and is rejected for the same reasons as Claim 11.

In regard to Claim 14, Lattner teaches that the Operator Class (Page 23) takes operands of type SimpleValue, and hence SimpleValue represents an Operand class.

In regard to Claim 15, the examiner takes official notice that abstract classes are an often used and beneficial programming concept, because abstract classes lets subclasses redefine the implementation of an interface while preserving the polymorphism of those classes. Claims 30, 40, and 48 correspond directly with Claim 15 and are rejected for the same reasons as Claim 15.

In regard to Claim 16, Lattner teaches an Operator class (Page 23) that takes operands of type SimpleValue, and hence SimpleValue represent an Operand class. Claims 31, 41, and 49 correspond directly with Claim 16 and are rejected for the same reasons as Claim 16.

In regard to Claim 17, Lattner teaches that operator classes correspond to different operations (i.e. Add, Subtract, Multiply). The class name is information relating to the specific operator and its function. The SimpleValue Class is obviously a class that stores a number, and hence would obviously contain information indicating a specific number. Claims 32, 42, and 50 correspond directly with Claim 17 and are rejected for the same reasons as Claim 17.

In regard to Claim 18, the examiner takes official notice that storing objects in object libraries is a well-known method of organizing classes for easier and more efficient access. Claims 33, 43, and 51 correspond directly with Claim 18 and are rejected for the same reasons as Claim 18.

In regard to Claim 19, Wang and Lattner teach the method of Claim 11, and Wang teaches that a VisualBasic Script interpreter performs standard VisualBasic Script interpretation (Column 6, lines 9-13). Program interpretation inherently includes steps that perform instruction translation and hence modification. In this way, modifications of instructions are required in said executing the program language representation in an interpreter. Since an interpreter does not change the functionality of the program, the modified program will produce the same results. Wang does not teach that platform-independent programming language objects are modified, however, since Lattner does teach objects that represent instructions, it would be obvious to pass instruction objects in an interpreter, and modify the instruction objects (again, an inherent aspect of the interpreter), instead of the instructions themselves. Claims 34 and 52 corresponds directly with Claim 19 and are rejected for the same reasons as Claim 19.

In regard to Claims 21 and 23, the examiner takes official notice that removing methods and fields from a program object is well known, especially when methods and fields become outdated, and hence no longer have any function.

In regard to Claim 24, Wang teaches: detecting one or more script language instructions in a markup language document by a Web browser (Column 4, lines 43-47).

In regard to Claim 25, Wang teaches a Java-based web browser executing within a Java Virtual Machine that executes the HTML parser (Column 2, lines 49-63). Claims 36, 44, and 53 correspond directly with Claim 25 and are rejected for the same reasons as Claim 25.

In regard to Claim 26, Wang teaches that the interpreter engine generates the platform-independent programming language representation (Column 7, lines 4-12).

In regard to Claim 27, Wang teaches that the platform-independent language is Java (Column 7, lines 4-8). Wang teaches that JavaScript is a well-known scripting language, and that the translation of JavaScript in a Web environment is also known in the art (Column 1, lines 13-18). Claims 37, 45, and 54 correspond directly with Claim 27 and are rejected for the same reasons as Claim 27.

In regard to Claim 28, Wang teaches: (a) a web server that inherently has a processor (Figure 1, item 106); (b) and a memory for storing instructions; (c) accessing a sequence of script language instructions (Column 4, lines 43-47); (d) generating a platform-independent representation of the one or more script language instructions (Column 4, lines 48-58); (e) interpret and execute the platform-independent programming language so that it produces results in accordance with the original sequence of script language instructions. The interpreter inherently converts the script into an equivalent intermediate form (Column 1, lines 19-21). Wang teaches that the platform-independent programming language representation comprises a sequence of logical commands (an interpreter naturally translates the instructions into an interpreted language) but does not teach that each of the commands is stored as one or more platform-independent programming language objects. Lattner, however, does teach representing instructions using a Java 'Instruction' class (Page 2, lines 18-26). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to access a sequence of script language instructions and generate a platform-independent representation of the one or more script language instructions where the platform independent representation produces equivalent results with the script language instructions, as taught by Wang, where the platform-independent representation includes instructions represented by objects, as taught by Lattner,

since programming objects are easier to create and manipulate. Claim 38 corresponds with Claim 28 and is rejected for the same reasons as Claim 28.

In regard to Claim 35, Wang teaches: (a) detecting script instructions in a markup language document; (b) passing execution to an interpreter engine which then accesses the instructions (Column 4, lines 43-58).

6. Claim 12, 29, 39, and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang (U.S. Patent Number 6,292,936) in view of "The Principles of Computer Hardware, Third Edition" by Alan Clements, 2000 (hereinafter Clements).

In regard to Claim 12, Wang teaches the method of Claim 11, but does not teach where the instructions are stored on a stack, and the instructions are popped from the stack during the executing step. Clements, however, does teach using the stack data structure to hold instructions that are executed by popping the instruction off of the stack. Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the detecting, generating, interpreting, executing, and accessing steps of Claim 11, as taught by Wang, where the instructions are stored on a stack, and the instructions are popped from the stack during the interpreting and executing steps, as taught by Clements, since this is an intuitive way to parse instructions in a computer system. Claims 29, 39, and 47 correspond directly with Claim 12 and are rejected for the same reasons as Claim 12.

7. Claims 20 and 22 are rejected under 35 U.S.C. 103(a) as being unpatentable over Wang (U.S. Patent Number 6,292,936) in view of "The IR to VMx86 Translation Module Specification" by Chris Lattner, December 1999 (hereinafter Lattner) and further in view of "Load-time Structural Reflection in Java" by Shigeru Chiba, June 2000 (hereinafter Chiba).

In regard to Claim 20, Wang and Lattner teach the method of Claim 19, but do not teach modifying programming language objects by adding one or more methods to the objects. Chiba, however, does teach using the Java Reflection API to alter class definitions and specifically, adding methods to a program object (Page 8, table 3). Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify programming language objects in order to generate a programming language that produces the same results as the first programming language, as taught by Wang and Lattner, where the modification includes adding methods to an object, since this allows for efficient dynamic alteration of Java classes.

In regard to Claim 22, Wang and Lattner teach the method of Claim 19, but do not teach modifying programming language objects by adding one or more fields to the objects. Chiba, however, does teach using the Java Reflection API to alter class definitions and specifically, adding methods to a program object (Page 8, table 3). Therefore it would have been obvious to one of ordinary skill in the art at the time of the invention to modify programming language objects in order to generate a programming language that produces the same results as the first programming language, as taught by Wang and Lattner, where the modification includes adding fields to an object, since this allows for efficient dynamic alteration of Java classes.

Conclusion

8. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

Renshaw (U.S. Patent Number 6,065,024)

Toutonghi et al. (U.S. Patent Number 5,920,720)

"JDK 1.1.8 Documentation: Java Reflection", Sun Microsystems, 1998

"Take an in-depth look at the Java Reflection API", by Chuck McManis, JavaWorld (www.javaworld.com), Issue September 1997.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth A Gross whose telephone number is (703) 305-0542. The examiner can normally be reached on Mon-Fri 7:30-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on (703) 305-4552. The fax phone numbers for the organization where this application or proceeding is assigned are (703) 746-7239 for regular communications and (703) 746-7240 for After Final communications.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

KAG
July 24, 2003

TUAN Q. DAM
PRIMARY EXAMINER